

Multi-Field Structural Decomposition for Question Answering

Tomasz Jurczyk

Mathematics and Computer Science
Emory University
Atlanta, GA 30322, USA
tomasz.jurczyk@emory.edu

Jinho D. Choi

Mathematics and Computer Science
Emory University
Atlanta, GA 30322, USA
jinho.choi@emory.edu

Abstract

This paper presents a precursory yet novel approach to the question answering task using structural decomposition. Our system first generates linguistic structures such as syntactic and semantic trees from text, decomposes them into multiple fields, then indexes the terms in each field. For each question, it decomposes the question into multiple fields, measures the relevance score of each field to the indexed ones, then ranks all documents by their relevance scores and weights associated with the fields, where the weights are learned through statistical modeling. Our final model gives an absolute improvement of over 40% to the baseline approach using simple search for detecting documents containing answers.

1 Introduction

Towards machine reading, question answering has recently gained lots of interest among researchers from both natural language processing (Moschitti and Quarteroni, 2011; Yih et al., 2013; Hixon et al., 2015) and information retrieval (Schiffman et al., 2007; Kolomiyets and Moens, 2011). People from these two research fields, NLP and IR, have shown tremendous progress on question answering, yet only few efforts have been made to adapt technologies from both sides. The NLP side often tackles the task by analyzing linguistic aspects, whereas the IR side tackles it by searching likely patterns.

While these two approaches perform well individually, more sophisticated solutions are needed to handle a wide range of questions. By considering linguistic structures such as syntactic and semantic trees, QA systems can infer deeper meaning of the context and handle more complex questions. However, extracting answers from these structures through either graph matching or predicate logic is

not necessarily scalable when the size of the context is large. On the other hand, searching patterns is scalable for large data, especially when coupled with indexing, although it does not always concern with the actual meaning of the context.

We present a multi-field weighted indexing approach for question answering that combines good aspects of both NLP and IR. We begin by describing how linguistic structures are decomposed into multiple fields (Section 3.3), and explain how the decomposed fields are used to rank documents containing answers through statistical learning (Sections 3.4 and 3.5). We evaluate our approach to 8 types of questions; our final model shows significant improvement over the baseline model using simple search (Section 4).

2 Related Work

Shen and Lapata (2007) assessed the contribution of semantic roles to factoid question answering and showed promising results. Pizzato and Mollá (2008) proposed a question prediction language model providing rich information and achieved improved speed and accuracy. Although related, our work is distinguished from theirs because we consider multiple fields whereas the others consider only one field representing semantic roles. Ferrucci et al. (2010) presented IBM Watson taking a hybrid approach between NLP and IR, and advanced the question answering task to another level.

Fader et al. (2013) proposed a paraphrase-driven perceptron learning approach using a seed lexicon. Our learning process is similar; however, it is distinguished in a way that we learn weights for individual fields instead of lexicons. Yih et al. (2014) introduced a semantic parsing framework for open domain question answering, which used convolutional neural networks for measuring similarities between decomposed entities. Weston et al. (2015) presented the Memory Networks models designed to memorize information about known objects and

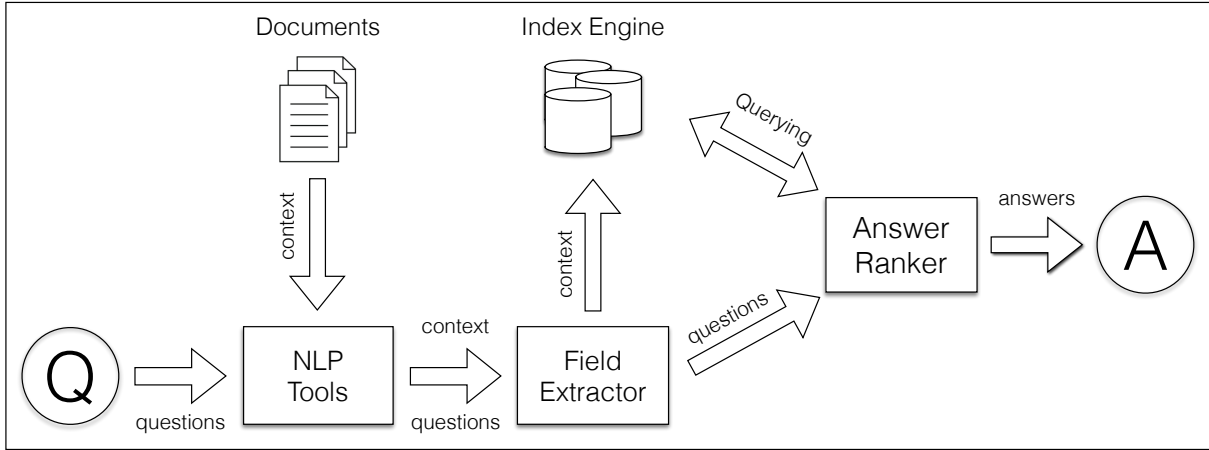


Figure 1: The overall framework of our question answering system.

actors. Our work is related to the this work; however, memory networks are designed to store and manipulate information about specific types of objects while our framework is generalizable to any type of objects induced from the context.

3 Approach

3.1 Overall framework

Figure 1 shows the overall framework. Our system is designed in a modular architectural way, so any further extension of fields can be easily integrated. The system takes input documents, generates linguistic structures using NLP tools, decomposes them into multiple fields, and indexes those fields. Questions are processed in the same way. To answer a question, the system queries the index for each field extracted from the question and measures the relevance score. All documents are ranked with respect to the relevance scores and their weights associated with the fields, and the document with the highest score is selected as the answer.

3.2 Modules

Our system consists of several modules closely connected together providing a fully working solution for the question answering selection task.

3.2.1 Documents and questions

Documents provide the context where the questions find their answers from. Each document can contain one or more sentences, in which answers for coming questions are annotated for training. Documents may simply be Wikipedia articles, news articles, fictional stories, etc. Questions are treated as regular documents containing only one sentence.

3.2.2 NLP tools

For the generation of syntactic and semantic structures, we used the part-of-speech tagger (Choi and Palmer, 2012), the dependency parser (Choi and McCallum, 2013), the semantic role labeler (Choi and Palmer, 2011), and the coreference resolution tool in ClearNLP¹. Ensuring good and robust accuracy for these NLP tools is important because all the following modules depend on their output.

3.2.3 Field extractor

The field extractor takes the linguistic structures from the NLP tools and decomposes them into multiple fields (Section 3.3). All fields extracted from the documents are passed to the index engine, whereas fields extracted from the questions are sent directly to the answer ranker module.

3.2.4 Index engine

The index engine is a search server that receives a list of fields decomposed by the field extractor, indexes terms in the fields, and responses to the queries generated from questions with their relevance scores. We used Elastic Search², as it provides a distributed, multi-tenancy-capable search.

3.2.5 Answer ranker

The answer ranker takes the decomposed fields extracted from a question, converts them into queries, and builds a matrix of documents with their relevance scores across all fields through the index engine (Section 3.4). It also uses different weights for individual fields trained by statistical modeling (Section 3.5).

¹<http://www.clearnlp.com>

²<https://www.elastic.co>

3.3 Structural decomposition

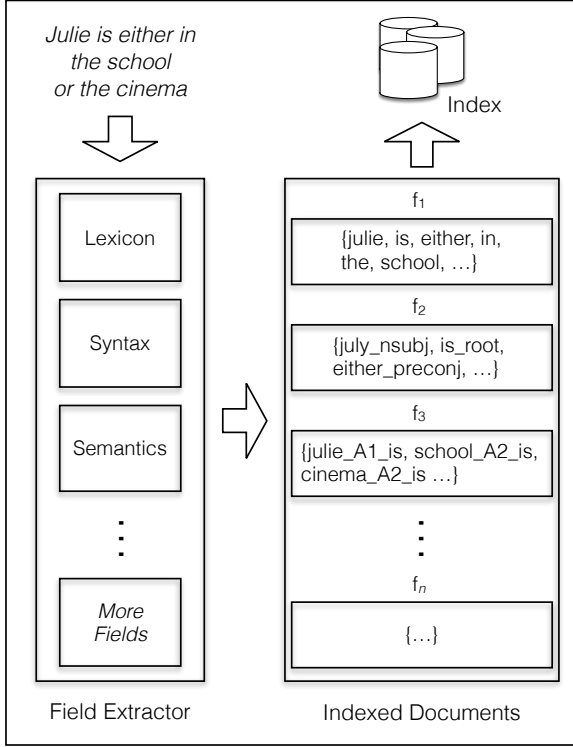


Figure 2: The flow of the sentence, *Julie is either in the school or the cinema*, through our system.

Each sentence is represented by the index engine as a document with multiple fields grouped into categories. Figure 2 shows an example of how the sentence is decomposed into multiple fields consisting of syntactic and semantic structures. Due to the extensible nature of our field extractor, additional groups and fields can be easily integrated. Currently, our system supports 24 fields grouped into the following three categories:

- Lexical fields (e.g., word-forms, lemmas).
- Syntactic fields (e.g., dependency labels).
- Semantic fields (e.g., semantic roles, distances between predicates).

3.4 Answer ranking

When a question q is asked, it is decomposed into the n -number of fields. Each field is transformed into a query where certain words are replaced with wildcards (e.g., $\{where_a1, is_pred, she_a2\} \rightarrow \{*_a1 is_pred she_a2\}$). Then, the relevance score r is measured between each field in the question and the same field in each document $d^t \in D$ by the index engine.³ The product of the relevance scores

³We set the elastic search results limit to 20.

and individual weights for all fields are summed, and the document \hat{d} with the highest score f is taken as the answer. Note that in our dataset, each document contains only one sentence so that retrieving a document is equivalent to retrieving a sentence. The following equations describe how the document \hat{d} is selected by measuring the overall score $f(q, d^t)$ using the relevance scores $r(q_i, d_i^t)$ and the weights λ_i .

$$\hat{d} = \arg \max_{d^t \in D} f(q, d^t)$$

$$f(q, d^t) = \sum_{i=1}^n \lambda_i \cdot r(q_i, d_i^t)$$

$$r(q_i, d_i^t) = \sum_{v \in q_i \cap d_i^t} \text{tf}_i^t(v) \cdot \text{idf}_i(v) \cdot \text{norm}_i^t(v)$$

3.5 Training weights for individual fields

Algorithm 1 shows how the weights for all fields are learned during training. We adapt the averaged perceptron algorithm, which has been widely used for many NLP tasks. All the weights $\vec{\lambda}$ are initialized to 1. For each question $q \in Q$, it predicts the document \hat{d} that most likely contains the answer. If \hat{d} is incorrect, then it compares the relevance score r between (q, \hat{d}) and (q, d) for each field, and updates the weight accordingly, where d is the true document from the oracle. This procedure is repeated multiple times through iterations. Finally, the algorithm returns the averaged weights, where each dimension represents the weight for each field.

Algorithm 1 Averaged perceptron training.

Input: D : document set, Q : question set.
 M : max-number of iterations, α : learning rate.
Output: The averaged weight vector.

```

1:  $\vec{\lambda} \leftarrow 1$ ;  $\vec{\lambda}' \leftarrow 0$ 
2: for  $iter \in [1, M]$  do
3:   foreach  $q \in Q$  do
4:      $\hat{d} = \arg \max_{d^t \in D} f(q, d^t)$ 
5:     if  $\hat{d} \neq d$  then #  $d$  is the oracle
6:       foreach  $i \in [1, n]$  do # for each field
7:          $\delta \leftarrow \alpha \cdot \text{sign}[r(q_i, d_i) - r(q_i, \hat{d}_i)]$ 
8:          $\lambda_i \leftarrow \lambda_i + \delta$ 
9:        $\vec{\lambda}' \leftarrow \vec{\lambda}' + \vec{\lambda}$ 
10: return  $\vec{\lambda}' \cdot \frac{1}{M \cdot |Q|}$ 
```

All hyper-parameters were optimized on the development sets and evaluated on the test sets. For our experiments, we used the following hyper-parameters: $M = 40, \alpha = 0.002$.

Type	Lexical				Lexical + Syntax				Lexical + Syntax + Semantics			
	$\lambda = 1$		λ is learned		$\lambda = 1$		λ is learned		$\lambda = 1$		λ is learned	
	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR
1 (qa1)	39.62	61.73	39.62	61.73	29.90	48.05	40.50	61.47	72.60	85.07	100.0	100.0
2 (qa4)	62.90	81.45	62.90	81.45	64.00	82.00	64.00	82.00	55.70	77.85	64.10	82.05
3 (qa5)	37.10	54.00	38.20	54.70	48.00	62.15	48.40	62.25	72.60	82.65	94.20	96.33
4 (qa6)	64.00	75.07	64.00	75.07	65.80	78.47	66.10	78.53	78.20	88.33	89.30	94.27
5 (qa9)	47.90	63.50	48.10	63.62	47.90	63.67	50.50	65.47	53.90	67.88	94.40	96.72
6 (qa10)	47.80	63.78	47.90	63.92	49.20	65.52	50.20	66.33	57.60	70.68	96.90	98.23
7 (qa12)	19.20	38.68	19.20	38.68	25.10	40.83	31.90	49.82	55.00	70.60	99.60	99.80
8 (qa20)	37.10	51.82	37.10	51.82	31.40	42.00	35.70	44.22	31.20	46.50	42.80	56.32
Avg.	44.45	61.25	44.63	61.37	45.16	60.34	48.41	63.76	59.60	73.70	85.16	90.47

Table 1: Results from our question-answering system on 8 types of questions in the bAbI tasks.

4 Experiments

4.1 Data and evaluation metrics

Our approach is evaluated on a subset of the bAbI tasks (Weston et al., 2015). The original data contains 20 tasks, where each task represents a different kind of question answering challenge. We select 8 tasks, in which answer for a single question is located within a single sentence. For consistency and replicability, we follow the same training, development, and evaluation set splits as provided, where every set contains 1,000 questions.

For the evaluation metrics, we use mean average precision (MAP) and mean reciprocal rank (MRR) of the top-3 predictions. The mean average precision is measured by counting the number of questions, for which sentences containing the answers are correctly selected as the best predictions. The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer. Mean reciprocal rank is the average of the reciprocal ranks of all question queries.

4.2 Evaluation

Table 1 shows the results from our system on different types of questions. The MAP and MRR show clear correlation with respect to the number of active fields. For the majority of tasks, using only the lexical fields does not perform well. The fictional stories included in this data often contain multiple occurrences of the same lexicons, and the lexical fields alone are not able to select the correct answer. Significantly lower accuracy for the last task is due to a fact that besides an answer is located within a single sentence, multiple passages for the single question are required to correctly locate the sentence with the answers. Lexical fields coupled with only syntactic fields do not perform much better. It may be due to a fact that the syntactic fields con-

taining ordinary dependency labels do not provide sufficient context-wise information so that they do not generate enough features for statistical learning to capture specific characteristic of the context. The significant improvement, however, is reached when the semantics fields are added as they provide deeper understanding of the context.

Not that this data set has also been used for evaluating the Memory Networks approach to question answering (Weston et al., 2015). The authors achieved high accuracy, reaching 100% in several tasks; however, our work still finds its own value because our approach is completely data-driven such that it can be easily adapted or extended to other types of questions. As a matter of fact, we are using the same system for all tasks with different trained models, yet still able to achieve high accuracy for most tasks we evaluate on.

5 Conclusion

This paper presents a multi-field weighted indexing approach for question answering. Our system decomposes linguistic structures into multiple fields, indexes terms of individual fields, and retrieves the documents containing the answers with respect to the relevance scores weighted differently. We observe significant improvement as we add more semantic fields and apply averaged perceptron learning to statistically designate weights for the fields.

In the future, we plan to extend our work by integrating additional layers of fields (e.g., Freebase, WordNet). Furthermore, we plan to improve our NLP tools to enable even deeper understanding of the context for more complex question answering.

References

- Jinho D. Choi and Andrew McCallum. 2013. Transition-based Dependency Parsing with Selec-

- tional Branching. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL'13, pages 1052–1062, Sofia, Bulgaria, August.
- Jinho D. Choi and Martha Palmer. 2011. Transition-based Semantic Role Labeling Using Predicate Argument Clustering. In *Proceedings of ACL workshop on Relational Models of Semantics*, RELMS'11, pages 37–45.
- Jinho D. Choi and Martha Palmer. 2012. Fast and Robust Part-of-Speech Tagging Using Dynamic Model Selection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL'12, pages 363–367, Jeju Island, Korea, July.
- Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-Driven Learning for Open Question Answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, ACL'13, pages 1608–1618.
- David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. 2010. Building Watson: An Overview of the DeepQA Project. *AI Magazine*, 31(3):59–79.
- Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. 2015. Learning Knowledge Graphs for Question Answering through Conversational Dialog. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL'15, pages 851–861.
- Oleksandr Kolomiyets and Marie-Francine Moens. 2011. A Survey on Question Answering Technology from an Information Retrieval Perspective. *Information Sciences*, 181(24):5412–5434.
- Alessandro Moschitti and Silvia Quarteroni. 2011. Linguistic kernels for answer re-ranking in question answering systems. *Information and Processing Management*, 47(6):825–842.
- Luiz Augusto Pizzato and Diego Mollá. 2008. Indexing on Semantic Roles for Question Answering. In *Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, IR4QA'08, pages 74–81.
- Barry Schiffman, Kathleen McKeown, Ralph Grishman, and James Allan. 2007. Question Answering Using Integrated Information Retrieval and Information Extraction. In *The Conference of the North American Chapter of the Association for Computational Linguistics*, ACL'07, pages 532–539.
- Dan Shen and Mirella Lapata. 2007. Using Semantic Roles to Improve Question Answering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL'07, pages 12–21.
- Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. *arXiv:1502.05698*.
- Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question Answering Using Enhanced Lexical Semantic Models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, ACL'13, pages 1744–1753.
- Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic Parsing for Single-Relation Question Answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL'14, pages 643–648.